The Role of a Descriptive Markup Language in the Creation of Interactive Multimedia Documents for Customized Electronic Delivery

Gil C. Cruz and Thomas H. Judd

Multimedia Communications Research Division
Bellcore
Morristown, N.J., USA 07960-1910

ABSTRACT: The emerging broadband telecommunications network promises to support a myriad of new mass-market information services that may in turn create a tremendous demand for new source material capable of exploiting the multimedia transport capability of the network. Authoring such material is presently a complex and time consuming process requiring specialized tools. We propose that a descriptive markup language, based on SGML and enhanced for interactive multimedia applications, can form the basis for a new set of authoring tools that will let experienced text authors transfer their skills to multimedia documents. Experience with a prototype version of such a language in the production of an experimental electronic magazine indicates that the approach is valid and useful. Future work includes defining text-like structure in temporal media and creating a unified set of editing and previewing tools.

KEYWORDS: authoring, hypermedia, interactive, markup, SGML

1 Introduction

The emerging digital telecommunications network will provide both increased bandwidth and enhanced flexibility, simultaneously transporting information streams in multiple formats. While such a network promises to support innovative new multimedia services, vendors of these services may be hard pressed to meet the resulting demands for new material. On the other hand, the sheer volume of information that may be created in response to these demands could overwhelm potential users unless some means for screening or selective delivery can be provided. As a part of our research aimed at understanding the needs and requirements of such a future network, we

have focused on mechanisms for creating information packages we call *multimedia documents* and on the problems encountered while trying to ensure that users get only those documents that they need and want.

The same growing computer power that will control the increasingly intelligent telecommunications network could be used by service vendors to customize information streams, raising the possibility of a new type of service that delivers a customized package of information to each individual subscriber. Information in the form of multimedia documents incorporating text, graphics, audio, and still or moving images could be accessed interactively on an electronic terminal such as a high-definition television (HDTV) set enhanced with an electronic pointing device. The future network could route these documents through *customizing agents* that provide information filters, distributing particular documents only to users who want and need them.

Much R&D in networking, information filtering, and display will be needed before such information services will be possible. Even if the delivery problems are solved, however, the problem of creating appropriate multimedia documents will remain. Historically, interactive multimedia presentations have required a complex and time consuming authoring process involving a team of specialists using media-specific tools, scripts, storyboards, and computer programming languages. While this may be inevitable for complex presentations, it places too many demands on the authors who will labor to meet the demands of mass-market information services.

1.1 Existing Media-Specific Tools

Various sophisticated tools have been constructed to aid the authoring process [Hodges *et al.* 1989; Kajimoto *et al.* 1989; Tabuchi & Muracho 1989; Mackay & Davenport 1989]. While they are a step in the right direction, these tools require authors to deal with specialized hardware and with different editors for different media, and they treat different media at different levels of abstraction. Text, for example, is manipulated by sentences or paragraphs, while video is edited frame-by-frame and audio millisecond-by-millisecond. Some recent video and audio editors deal at a higher level (e.g., the video scene) but all are specialized devices intended for specific purposes.

Specialized authoring systems to handle the task of unifying various media segments for presentation [Frenkel 1989] are typically intended for (and execute only on) a specific hardware configuration. Thus while

these authoring systems may help in the production of interactive programs, they contribute to the proliferation of proprietary delivery systems. The resulting lack of standardization has hindered the development of interactive systems and has made it difficult to reuse material among applications. Some system vendors have attempted to develop software that permits a degree of hardware independence among authoring systems or emulates the operation of one system while running on another platform [Magel 1989]. Such software may strengthen the position of de facto hardware standards, but it begs the question of true hardware and software independence.

1.2 The Goal of Unified Systems

One goal of our research has been to find better ways to create interactive multimedia documents, particularly ways that are independent of the characteristics of specific display devices or delivery systems. We want authors to work at a comfortable level of abstraction, ideally treating all media at the same level with the same tools. Such unified authoring tools might provide an evolutionary path for authors to transfer their text skills to the creation of interactive multimedia documents.

A key element in the creation of such unified authoring systems is a high-level document markup language that will permit an author to deal with the *structure* of a document independent of its *presentation*. If such a language can handle both the spatial structure of text and the temporal structure of audio/video, then an author could treat all media at the same level of abstraction. If all its elements are described in a complete and unambiguous way, a document can be formatted and presented automatically by computer programs that resolve the details of differing hardware systems.

Our research indicates that such a markup language is both possible and practical. In Section 2 of this paper, we present the concept of markup languages, discuss the Standard Generalized Markup Language (SGML), and present our prototype SGML-based markup language with its extensions for interactive multimedia documents. Section 3 describes an experimental application of our language—the creation of interactive multimedia articles for a prototype electronic magazine. Section 4 discusses related work in this field and outlines some thoughts for future work.

2 Markup Languages

Although the term *markup* originally referred to the process of marking manuscript copy for manual typesetting [Chicago 1982], it now generally refers to instructions included in a document to control machine processing. One comprehensive review [Coombs *et al.* 1987] identified six types of markup used in preparing text documents. Of the six, only two types are directly applicable to machine processing of text.

Procedural markup consists of commands indicating how text should be formatted by a specific computer program; it must be rewritten when the text is processed by a different formatter. *Descriptive* markup inserts tags to identify the various text elements (e.g., <para> for paragraph). This markup describes what the text elements are, rather than how a particular text formatter should handle them. Formatting is a separate operation, so the structure of a document (defined by tags) can be considered separately from its appearance (defined by formatter rules).

Coombs, *et al.*, concluded that descriptive markup best supports the process of document development and publication, for a number of reasons:

- Authors can focus on content and structure rather than on presentation.
- Composition assistance procedures such as alternative views (display an outline, etc.) or structure-oriented editing (delete a footnote, etc.) can be used.
- Style changes can be made without exposing document contents to corruption.
- Separation of content from presentation encourages standards that support document portability.

The first two points address document creation for mass-market information services. The others are significant for documents delivered to differing terminals on a network where the presentation will be determined by service vendors rather than by authors.

2.1 The Standard Generalized Markup Language

SGML, the *Standard Generalized Markup Language* [ISO 1986], is an international standard meta-language defining the syntax of a descriptive

markup language [Barron 1989] that identifies the structural elements of a document with descriptive *tags,* each consisting of a delimited *generic identifier* (GI) with optional attributes. SGML specifies a markup syntax, not a vocabulary of specific GIs for the tags. The vocabulary of GIs and the rules for using them are described in an SGML construct known as a *document type definition* (DTD).

An SGML document is always interpreted according to a DTD that rigorously specifies the structure of the document and the element-tagging syntax. Authors can either create their own DTD or use an existing one such as the Association of American Publishers DTD for articles [AAP 1987]. Our research began with this DTD, then extended it to encompass the interactive multimedia nature of electronic documents.

Once the DTD is defined, an author can create a new document with a word processing system or with SGML-specific authoring software. The document is then processed by an SGML parser that checks for conformance with the DTD, performs tag expansions, and inserts implied tags. The parser creates an SGML-tagged document that is passed to a formatter that maps its elements onto a particular output device [Barron 1989]. The formatter operates according to a style definition that controls the appearance of its output. This definition might be represented in the Document Style Semantics and Specification Language (DSSSL) [ISO 1988] in the form of a set of rules to be adhered to in presenting a document.

There are other descriptive markup languages, each with its cadre of supporters. However, SGML is an international standard with a growing number of users. In 1987 the U.S. Department of Defense promulgated MIL-STD-1840A [White 1989], commonly known as the Computer-aided Acquisition and Logistic Support (CALS) program, requiring all defense contractors to use the military version of SGML (MIL-M-28001) for their documentation. In 1989 Federal Information Processing Standard (FIPS) 152 required that all hardware and software procurements related to text and document processing be capable of handling SGML. As with many government decrees, these may well result in SGML becoming **the** standard for document preparation, regardless of the merits of competing systems.

2.2 Extensions for Interactive Multimedia Documents

We have created a research prototype SGML markup language based

on the AAP's DTD for printed articles. We added tags for temporal media elements (audio and video) and a mechanism to link these elements to text, thus providing for control of non-text media on an interactive electronic display terminal.

With our language, text defines the basal structure of an electronic document. Other media are introduced through *sidebars,* a generalized concept for constructs that interrupt the linear presentation of a document. The language provides a mechanism for creating *links* from text elements to sidebar elements defined in a separate section of the document. These links allow for interaction with the document—the "from" element in the text indicates that the link exists, while the "to" element specifies the resulting action if the user chooses to follow that link. An example of a sidebar is a timed sequence of images (with optional caption and audio) called a *slideshow.* It is the "to" element of a link and is activated when a reader triggers the "from" element, a specified text string in the body of the document.

A sidebar is created by referencing it in the body of an article, then defining it in a sidebar definition section. For example, an author might include the following in the body of an article to create a slideshow sidebar that illustrates a point:

```
<body>
 ... to <sidebar_ref [id=startup]> start the machine
          </sidebar_ref> one must first ...
          :
</body>
```

followed by this after the end of the body:

```
<sidebar_def [id=startup]>
<slideshow [type=ramp]>
<caption> Start-Up Sequence </caption>
<audio_ref [id=sound_effect1]>
<slide [id=slide1; duration=4]>
<slide [id=slide2; duration=2]>
<slide [id=slide3; duration=3]>
</slideshow>
</sidebar_def>
```

Depending on the style definition in effect at the time, a formatter might represent the "from" element of the link by highlighting the string "start the machine" in the body of the document. If the user

"clicked" this highlighted string, the slideshow would be triggered; a window would pop up, sequentially displaying the images with audio and a caption. In this example, image *slide1* would appear for 4 seconds, followed by *slide2* for 2 seconds and finally *slide3* for 3 seconds. The slideshow was declared to be of type *ramp* so its images would be displayed in a diagonal array. This example shows how an author could create a slideshow by describing its spatial and temporal structure and declaring a sidebar link from within running text. Similar constructs handle audio and video clips.

If documents are to be distributed via customizing agents with information filters, those agents must be able to ascertain their contents. Descriptive markup aids this process by explicitly identifying document elements. It is straightforward, for example, to distinguish a keyword in a title from one in a paragraph, allowing context-dependent keyword matching. However, it is difficult or impossible to surmise an author's intent in matters like the degree of expertise, prerequisite knowledge, or demographics of the intended reader. Thus our prototype markup language allows an author or editor to embed explicit classification information in the source file.

If this paper were to be prepared with the prototype language, it might include *class tags* such as:

```
<class [field=computers]>
<class [area=publishing]>
<class [aspect=interactive]>
<class [aspect=multimedia]>
<class [occupation=professional]>
<class [breadth=2]>
<class [expertise=4]>
<class [prerequisite=4]>
<class [education=4]>
```

The first five tags, for which a glossary of keywords must be provided, indicate that this is a paper on computer-based publishing of interactive multimedia documents and that it was intended for professionals in the field. The other tags, with metric values on a scale of 1 to 5, indicate that the subject matter is fairly narrowly focused and is intended for users with considerable expertise, prerequisite knowledge, and a college education. Information conveyed by the first four tags could be inferred from the text, but that in the last five tags would be difficult if not impossible to extract from the text alone.

3 A Specific Instance: The Electronic Magazine

As an experiment to test our prototype markup language, we have been investigating the concept of an information service that could use the future telecommunications network to distribute information in the format of an electronic magazine containing interactive multimedia documents called articles [Judd & Cruz 1989]. The user controls viewing of these articles; two readers of the same article might access different elements in a different order.

Figure 1 shows a simple flow chart of the creation and delivery of such a magazine. Publishers (teams of authors and human editors) create articles and use the network to deposit them in a source database. A filter program matches characteristics of articles in the database against a profile of an individual user's interests. The filter selects appropriate articles from the database, packages them in a format similar to a printed magazine, creates a summary of the information, and delivers the customized package to the user's terminal.

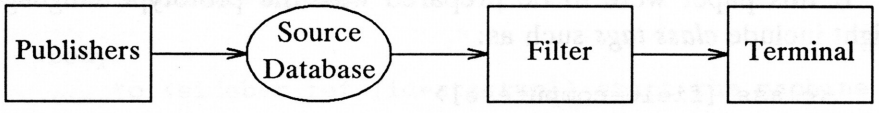

Figure 1: Electronic Magazine Flow Chart

To test the markup language, we created a set of software tools that formats, filters, and displays multimedia articles. Using these tools, we have created more than 90 articles with text, images, slideshows, audio segments, and video clips. Although our present filter has only rudimentary keyword matching capabilities, we have succeeded in automatically generating multimedia magazines for people with different interest profiles. This experience has strengthened our belief that a descriptive markup language can bring significant benefits to the creation and dissemination of multimedia articles. Specific advantages we have found include:

- A multimedia article can be created with a standard text editor, once the visual and audible elements have been created.
- Article elements are explicitly tagged so appropriate entries in the source database can be automatically created.
- Since an article's elements retain their tags in the database, it is straightforward for the filter to access those needed to create a summary of a magazine's contents.
- Because the presentation of an article is determined by a style definition (set of rules) belonging to the formatter, not to the source material, it is easy to change the appearance of an article.

4 Future & Related Work

Text presently defines the basal structure of a document; other media are introduced via links into sidebars. It may be possible to define a unique structure for each medium, and hence to create an arbitrary basal structure. We believe that there is a distinct advantage in adopting the structure of text to all media, however, at least in the particular case of authoring documents for mass-market services. Basing a multimedia editor on the familiar paradigm of a text editor could let authors apply a lifetime of acquired language skills to the less familiar world of audio and video communication. Our current work includes defining media structures that parallel that of text, and determining what computer tools might best support future authors.

4.1 Structure in Temporal Media

One possible text-related hierarchical structure for video might be:

video	text
program	document
scene	paragraph
shot	sentence
frame	word

Based on this model, a video program (document) is made up of scenes (paragraphs) containing shots (sentences) which are in turn comprised of frames (words). A text-like editor could then be used to rearrange scenes in video as easily as paragraphs in text.

Without such structure it is difficult to define links between arbitrary media; with it inter-media links could be declared freely. For example, the basal structure of a document might be defined by video, with other media introduced by sidebars. Using SGML notation, the source document could contain the following reference:

```
<video_program [id=vidprog]>
<video_scene [id=vid1]>
<sidebar_ref [id=other_medium]> <video_scene [id=vid2]>
</sidebar_ref>
<video_scene [id=vid3]>
</video_program>
```

If the reader "clicked" on the image while viewing scene *vid2* of program *vidprog,* the video would be suspended and sidebar *other_medium* would be activated. At its conclusion, *vid2* would resume, followed by *vid3.*

An audio program could be handled similarly. For speech, which can be viewed as another manifestation of text, the parallel structure would be nearly identical, given the addition of some sense of time. Although music also has structure, it may require more of a leap of faith to reduce it to a text-like representation. An effort has been made by the Music in Information Processing Standards (MIPS) committee of ANSI to apply SGML structure to the development of SMDL, the Standard Music Description Language [Newcomb & Goldfarb 1989]. The HyTime subset [Goldfarb & Talbot 1989] of that language contains a model for the duration and synchronization of events. Although developed for describing music, the SMDL time model is applicable to multimedia and time-sequenced documents in a more general sense; HyTime could be used as a language for the creation of such documents. Because of the differences in goals, SMDL and HyTime are quite different from our prototype markup language, with its strongly text-oriented approach. Our language is suited to the needs of text authors who wish to expand into multimedia documents. SMDL, on the other hand, has a sophisticated and flexible notion of time and synchronization that is presently missing from our language.

4.2 Unified Editor/Previewer Environment

Encoding the structure of all media with SGML tags provides the necessary "hooks" for the arbitrary inter-media links needed for true

multimedia documents. Beyond providing these multimedia hooks, encoding an explicit structure for all elements of a document could support the creation of a unified media editor. Text elements would be designated with conventional tags delineating text tokens represented by character strings. Temporal elements, on the other hand, would be designated by tags with generic identifiers reflecting the nature of the medium. Audio or video tokens would be represented by surrogates, graphical icons or other symbols within the lexicon of the editing computer. Based on the text paradigm, such an editor could manipulate the various elements of a document much as text is now manipulated by a word processor. Since it can be created on the same system that manipulates it, text will always enjoy an inherent editing advantage; audio and video tokens must be created elsewhere and merely rearranged or truncated by the editing system. Nevertheless, such a unified editor could be a powerful tool in the hands of an author who needed to manipulate all the elements of a multimedia document.

To complete such a system, the author needs a previewer that displays output from the unified editor on a terminal. The previewer (a particular instance of a formatter) would take the document elements as delineated by SGML tags and map them into representations appropriate for a terminal. To do this, it would employ a set of rules that define the style and design of the presentation on a given terminal. By loading different sets of rules, the previewer could emulate different terminals. With such a previewer and unified media editor, an author could refine multimedia documents by iteratively editing and viewing them.

4.3 From Here to Hypermedia

We have been considering links between elements of self-contained interactive documents. More generally, links from one document could refer to elements of another, leading to *hypermedia* systems in which documents are specific instances of multimedia elements drawn from a database of information. Hypermedia is a generalization of hypertext, a database technology typically characterized by small fragments of text interconnected by machine-supported links [Conklin 1987].

In a hypermedia system, information elements would be encoded by descriptive markup, but the "chunks" of information would be smaller than the self-contained documents of the prototype electronic magazine. Authors would create individual elements in various media

and define links with attributes for external use. The user would determine the order of presentation, or *path* through a hypermedia document, by selecting links between elements. If link selection is unconstrained, the presentation may become confusing and disorienting to the user. One way to constrain link selection is to create a *script,* defined as "an active directed path through one or more documents that need not follow the linear order of the documents" [Zellweger 1988]. Rather than being a text document with links to other elements, an article might be a menu of scripts, each specifying a path through a collection of hypermedia elements. Each path might define a user experience that was the result of the efforts of multiple authors.

A descriptive markup language could be used in the preparation of such scripts. If they included our concept of class tags, the scripts could capture the author's intentions in writing them, and a customizing agent could select an appropriate script for an individual user. This procedure would preserve the flexibility and power of hypermedia systems, while ensuring that individuals received only suitable material.

5 Conclusion

The future broadband telecommunications network promises to support interactive multimedia presentations. For such presentations to become routine offerings of mass-market information service vendors, they must be much easier to create than present-day interactive video applications. Providing powerful authoring tools is an important step toward simplifying the creation of multimedia documents.

By explicitly defining the structure of a document, a descriptive markup language supports the creation of unified media editors and document previewers. Imposing the structure of text on temporal media such as audio or video permits the development of editing tools that are familiar and comfortable to authors of text documents. Such tools could help these authors transfer their skills to multimedia document production. SGML, an established international standard syntax with a growing list of applications, is a reasonable basis for such a markup language. Although most SGML applications have historically been for processing text, extensions to handle audio, video, and slideshows are straightforward and extend the benefits of descriptive markup to these media.

Documents created with descriptive markup are inherently suited

for processing by database-driven information filters and can be formatted for different output devices without changing source material. Extensions to the language allow the creation of links between elements of arbitrary media and ultimately allow the creation of full hypermedia links among arbitrary elements of an information space.

The research reported here is just the beginning of the work needed to realize the goal of unified tools to ease the creation of full hypermedia presentations. Our progress, along with the work of other researchers, indicates that SGML forms a solid basis for this work.

References

[AAP 1987] *Standard for Electronic Manuscript Preparation and Markup,* Electronic Markup Series, Association of American Publishers, Washington, D.C., August 1987.

[Barron 1989] David Barron, "Why use SGML?," *Electronic Publishing,* vol. 2, no. 1, April 1989, pp. 3-24.

[Chicago 1982] *The Chicago Manual of Style,* University of Chicago Press, 1982, 13th Edition.

[Conklin 1987] J. Conklin, "Hypertext: An Introduction and Survey," *IEEE Computer,* vol. 20, no. 9, September 1987, pp. 17-41.

[Coombs, *et al.* 1987] T. Coombs, A. Renear, and S. DeRose, "Markup Systems and the Future of Scholarly Text Processing," *Communications of the ACM*, vol. 30, no. 11, November 1987, pp. 933-947.

[Frenkel 1989] Karen A. Frenkel, "The Next Generation of Interactive Technologies," *Communications of the ACM,* vol. 32, no. 7, July 1989, pp. 872-881.

[Goldfarb & Talbot 1989] C. Goldfarb and A. Talbot, "Standard Music Description Language (SMDL) Part Two: Hypermedia/Time-Based Document Subset (HyTime)," *X3V1.8M/SD-6 Journal of Development,* Fifth draft, August 11, 1989.

[Hodges *et al.* 1989] M. Hodges, B. Davis, and R. Sasnett, "Investigations in Multimedia Design Documentation," in Edward Barrett (ed.), *The Society of Text,* MIT Press, 1989, pp. 79-89.

[ISO 1986] *Information Processing—Text and Office Systems—Standard Generalized Markup Language (SGML)* (ISO 8879), International Standards Organization, 1986.

[ISO 1988] *Information Processing—Text and Office Systems—Document Style Semantics and Specification Language (DSSSL),* International Standards Organization (ISO/IEC JTC 1/SC 18/WG 8 N606), July 1988.

[Judd & Cruz 1989] T. H. Judd and G. C. Cruz, "Customized Electronic Magazines: Electronic Publishing for Information Grazing," *Advanced Printing of Paper Summaries, Electronic Imaging '89,* October 1989, pp. 504-509.

[Kajimoto *et al.* 1989] K. Kajimoto, F. Nakayama, T. Nonomura, Y. Imai, S. Isoda, and Y. Kushiki, "New-Media Document (NewDoc) and Dynamic Navigation on the BTRON Specification," in *Digest of Papers,* COMPCON Spring '89, March 1989, pp. 40-42.

[Mackay & Davenport 1989] W. Mackay and G. Davenport, "Virtual Video Editing in Interactive Multimedia Applications," *Communications of the ACM,* vol. 32, no. 7, July 1989, pp. 802-810.

[Magel 1989] Mark Magel, "Springtime for System Integrators," *AV Video,* April 1989, pp. 48-53.

[Newcomb & Goldfarb 1989] S. Newcomb and C. Goldfarb, "Standard Music Description Language (SMDL) Part One: Objectives and Methodology," *X3V1.8M/SD-6 Journal of Development,* Fifth draft, August 11, 1989.

[Tabuchi & Muracho 1989] M. Tabuchi and Y. Muracho, "MeSOD: the Metric Spatial Object Data Model for a Multimedia Application: Hyperbook," in *Digest of Papers,* COMPCON Spring '89, March 1989, pp. 396-401.

[White 1989] William White, "'Til The CALS Come Home," *Electronic Publishing & Printing,* vol. 4, no. 9, December 1989, pp. 54-55.

[Zellweger 1988] Polle T. Zellweger, "Active Paths Through Multimedia Documents," in C. C. vanVliet (ed.), *Document Manipulation and Typography,* Proc. Int'l. Conference on Electronic Publishing, Document Manipulation and Typography, Cambridge University Press, 1988, pp. 19-34.